



500.43703X00

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

Applicant(s): Y. MASUOKA

Serial No.: 10/809,435

Filed: March 26, 2004

Title: METHOD FOR GENERATING POLICY RULES AND METHOD FOR  
CONTROLLING JOBS USING THE POLICY RULES

**LETTER CLAIMING RIGHT OF PRIORITY**

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

April 6, 2004

Sir:

Under the provisions of 35 USC 119 and 37 CFR 1.55, the applicant(s) hereby  
claim(s) the right of priority based on:

**Japanese Patent Application No. 2003-086919**  
**Filed: March 27, 2003**

A certified copy of said Japanese Patent Application is attached.

Respectfully submitted,

ANTONELLI, TERRY, STOUT & KRAUS, LLP

Paul J. Skwierawski

Registration No.: 32,173

PJS/rr  
Attachment

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日            2 0 0 3 年   3 月 2 7 日  
Date of Application:

出 願 番 号            特 願 2 0 0 3 - 0 8 6 9 1 9  
Application Number:  
[ST. 10/C]:            [ J P 2 0 0 3 - 0 8 6 9 1 9 ]

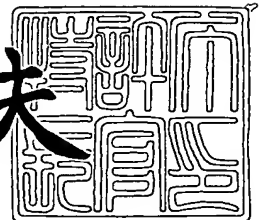
出   願   人            株 式 会 社 日 立 製 作 所  
Applicant(s):



2 0 0 4 年   3 月 1 8 日

特許庁長官  
Commissioner,  
Japan Patent Office

今 井 康 夫



出証番号   出証特 2 0 0 4 - 3 0 2 2 0 5 7

【書類名】 特許願

【整理番号】 H03002391A

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 17/60

【発明者】

    【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所中央研究所内

    【氏名】 増岡 義政

【特許出願人】

    【識別番号】 000005108

    【氏名又は名称】 株式会社 日立製作所

【代理人】

    【識別番号】 100075096

    【弁理士】

    【氏名又は名称】 作田 康夫

    【電話番号】 03-3212-1111

【手数料の表示】

    【予納台帳番号】 013088

    【納付金額】 21,000円

【提出物件の目録】

    【物件名】 明細書 1

    【物件名】 図面 1

    【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 ポリシールールの生成方法およびそれを用いたジョブ運用管理方法

【特許請求の範囲】

【請求項 1】

障害等のイベントが発生した場合のアクションを記載したポリシールールを用いて、情報処理システムにおいて、1つ以上のジョブの実行時の運用管理を自動実行するためのポリシールールの生成方法であって、

前記 1つ以上のジョブのそれぞれにつき、実行の制約条件と、該実行の制約条件を満たせない場合の損害額の評価方法を記載したジョブ実行条件を作成するステップと、

前記ジョブ実行条件を参照し、特定の前記ジョブにおいて、特定のイベントが発生した場合に、特定のアクションを取った時の損害額を評価するステップと、

前記損害額を評価するステップを、特定のジョブと、特定のイベントにおいて、複数準備されたアクションの全てに対して実行し、評価した損害額が最小になる場合のアクションを決定するステップ

を含むことを特徴とするポリシールール生成方法。

【請求項 2】

前記ジョブ実行条件を作成するステップは、前記実行の制約条件として、ジョブの要求終了時刻を記載するステップを含むことを特徴とする請求項 1 に記載のポリシールール生成方法。

【請求項 3】、

前記ジョブ実行条件を作成するステップは、前記制約条件を満たせない場合の損害額を評価する方法として関数を記載するステップを含み、

前記損害額を評価するステップは、前記関数を実行するステップを含むことを特徴とする請求項 1 に記載のポリシールール生成方法。

【請求項 4】

前記関数はジョブ終了の遅延時間を引数とする関数であり、前記関数を実行するステップではジョブ終了の遅延時間を計算することを特徴とする請求項 3 に記

載のポリシールール生成方法。

**【請求項 5】**

請求項 1 に記載のポリシールール生成方法において、  
前記 1 つ以上のジョブの終了予定時刻を含む、ジョブ実行スケジュールを作成するステップと、

1 つ以上の前記アクションのそれぞれを、前記ジョブ実行スケジュールを変更するサブルーチンとして記述するステップとをお更に有し、

前記特定のアクションを取った時の損害額を評価するステップは、当該アクションを記述したサブルーチンを実行し、前記ジョブ実行スケジュールを変更するステップ

を含むことを特徴とするのポリシールール生成方法。

**【請求項 6】**

前記評価した損害額が最小になる場合のアクションを決定するステップを、1 つ以上のジョブが実行されている予定の、すべての時刻において実行することを特徴とする請求項 1 に記載のポリシールール生成方法。

**【請求項 7】**

前記ポリシールールを用いて、情報処理システムにおいて、1 つ以上のジョブの運用管理を自動実行する方法であって、請求項 1 ないし請求項 6 に記載のポリシールール生成方法を含むことを特徴とするジョブ運用管理の自動実行方法。

**【発明の詳細な説明】**

**【0001】**

**【発明の属する技術分野】**

本発明は、情報処理システムに関し、特にネットワークを利用した複数の計算機を含む情報処理システムに関する。より具体的には、前記情報処理システムにおいて実行されるジョブの管理方法に関する。さらに具体的には、前記ジョブを実行している前記情報処理システムにおいて障害等のイベントが発生した時に、実行中のジョブを再実行するなどの運用管理を、ポリシールールを用いて自動実行する方法に関する。

**【0002】**

**【従来の技術】**

文献 Saldago 他、“System for managing resource deficient jobs in a multifunctional printing system”、米国特許 6,504,621 号に記載の情報処理システムには、ジョブの実行を管理するジョブマネージャにポリシールールを設定し、障害等のイベントが発生した場合の運用管理を自動実行する一方法が記載されている。

**【0003】**

当該文献によれば、情報処理システムにはジョブを管理する前記ジョブマネージャが設けられており、ジョブを実行させようとするユーザは、当該ジョブを前記ジョブマネージャに投入する時に、「当該ジョブの実行時に障害が発生した場合に取るアクション」をポリシールールとして指定する。アクションとしては、例えば「同じジョブを再実行する」「ユーザに連絡する」などが指定できる。前記ジョブマネージャは、情報処理システムにおいてハードウェア障害やソフトウェア障害が発生して、ジョブの実行が異常終了した場合には、前記ポリシールールを参照し、ユーザが指定したアクションを自動実行する。これにより、ジョブマネージャは、ジョブ実行中に障害等のイベントが発生した場合の運用管理を、自動実行することができる。

**【0004】****【非特許文献】**

Saldago 他、“System for managing resource deficient jobs in a multifunctional printing syst”

**【特許文献】**

米国特許 6,504,621 号

**【0005】****【発明が解決しようとする課題】**

前記の従来の技術は、次の点で問題がある。

**【0006】**

すなわち、ジョブごとに指定されるポリシールールが、他のジョブや障害等の発生時刻とは無関係に指定されているため、情報処理システムおよび当該情報処理システムを利用する企業にとって、最適な自動運用管理ができない場合がある。簡単な例を用いて具体的に説明する。

#### 【0007】

例として、ただ1台の計算機からなる情報処理システムを考える。当該情報処理システムには2つのジョブAおよびジョブBが投入されていて、ジョブAは当該計算機を占有して実行中であり、ジョブBは実行開始時刻をあらかじめ指定されており、その時刻になるまで実行を待っているとする。このとき、ジョブAのポリシーとして「ジョブ実行中に異常終了したら再実行する」と指定されているとすると、ジョブAは再実行され、それだけジョブAの終了時刻は遅れることになる。もしジョブAの終了時刻がジョブBの開始時刻よりも後になると、ジョブAが当該計算機を占有しているために、ジョブBは指定された開始時刻に開始できない。この場合、特にジョブBが、終了時刻を厳格に守らなくてはならないジョブだとする、ジョブAとジョブBの実行を担う当該情報処理システムとしては大きな問題となる。

#### 【0008】

逆に、後続のジョブBが重要であることに配慮して、先行して実行されるジョブAに「ジョブ実行中に異常終了したら、再実行しないで当該ジョブを破棄する」というポリシーが設定されているとする。もしジョブAの開始直後に障害が発生して、ジョブAが異常終了したとすると、ジョブAは実行されないことになる。ここで、もしジョブAを再起動すれば、ジョブBの開始時刻までにジョブAを終わることができたとする、当該情報処理システムとしては、ジョブAを実行しなかったことによって、計算機の時間を無駄にしまったことになる。

#### 【0009】

上記の例のように、従来の技術では、ポリシールールが他のジョブの状態（この例では、ジョブBの開始時間）や、障害等の発生時刻（この例では、ジョブAが異常終了した時刻）に無関係に指定されているため、情報処理システム全体として最適な自動運用管理ができない場合がある。上記の例ではジョブを実行する

計算機がただ1つで、投入するジョブは2つであったが、ジョブを実行できる計算機が多数あり、投入されたジョブも多数あるような情報処理システムでも、同じ問題が発生する。

#### 【0010】

そこで、本発明の第1の目的は、ジョブマネージャが情報処理システム全体として最適な自動運用管理をするための、前記ポリシーールの生成方法を提供することである。

#### 【0011】

本発明の第2の目的は、ジョブマネージャが、障害等のイベントの発生時刻やイベントの種類も加味して、情報処理システム全体として常に損害額が最小となるような、きめ細かくかつ最適な自動運用管理を行うための、前記ポリシーールの生成方法を提供することである。

#### 【0012】

##### 【課題を解決するための手段】

本発明においては、ジョブごとのポリシーールは、各ジョブの投入者ではなく、情報処理システムのシステム管理者によって、下記の方法により設定される。

まず、実行するジョブごとに、要求終了時刻と、終了時刻がその時刻を過ぎた場合、またはそのジョブを実行できなかった場合の損害額を設定する。また、その情報処理システムにおいて発生し得る障害の種類のリストと、取り得るアクションのリストを用意する。

これにより、ある時刻に、その時刻に実行中のあるジョブにおいて、ある障害が発生した時に、あるアクションを取った場合の、各ジョブの終了時刻を見積ることができる。さらに、見積った各ジョブの終了時刻を用いて、その場合の損害額の合計を計算することができる。

#### 【0013】

そこで、ある時刻（T）とその時刻に実行中のジョブ（J）と障害の種類（X）の組合せにおいて、取り得る各アクションに対して前記損害額の合計を計算し、当該損害額の合計が最も少ないアクション（A）を決定し、「時刻Tにおいて



、障害Xが発生したら、アクションAを実行する」というポリシールールを、ジョブJのポリシールールとして一つ決定する。同様の手続きを、時刻Tとその時刻に実行中のジョブJと障害の種類Xのあらゆる組合せに対して実行することにより、すべてのジョブの、すべての時刻における、すべての種類の障害に対するポリシールールの組が得られる。前記情報処理システムのシステム管理者は、得られたポリシールールの組を前記ジョブマネージャに設定する。

#### 【0014】

本発明の方法によれば、ジョブマネージャには、常に情報処理システム全体の損害額が最小になるようなポリシールールが設定される。従って、ジョブマネージャは、情報処理システム全体として最適な自動運用管理を行なうことができる。

#### 【0015】

##### 【発明の実施の形態】

以下、本発明の実施の一形態を、図面を参照しながら説明する。なお、簡単のため、本明細書中では「発明の実施の形態」を、単に「実施例」と呼ぶことにする。

##### (1) ポリシールール生成環境のハードウェア構成

本実施例においては、本発明によるポリシールールの生成は、情報処理システムのシステム管理者が、本発明のポリシールール生成方法を、ポリシールール生成計算機を用いたポリシールール生成環境において実行することにより行われる。図10を用いて、本実施例のポリシールール生成環境のハードウェア構成を説明する。ポリシールール生成環境150は、ポリシールール生成計算機151と、記憶装置153から構成される。ポリシールール生成計算機151は、実際には端末、パーソナルコンピュータ、メインフレームなどの計算機で、1つ以上のプロセッサを有し、ポリシールールを生成するための処理を実装したポリシー生成ツール152を実行する。なお、プロセッサについては図示していない。記憶装置153は磁気ディスク、メモリ、ファイルサーバ、ネットワーク情報サーバなどの記憶媒体であり、ジョブ実行スケジュール200、ジョブ実行条件リスト300、アクションリスト400、イベントリスト500、ポリシールールリス

ト 600 をデータとして格納することができる。記憶装置 153 は、ポリシールール生成計算機 151 と、リンク 154 によって接続されている。リンク 154 は、接続コードや、無線または有線のネットワークなどである。なお、図 10 では、ジョブ実行スケジュール 200、ジョブ実行条件リスト 300、アクションリスト 400、イベントリスト 500、ポリシールールリスト 600 はすべて同じ記憶装置 153 に格納されているが、実際には別々の記憶装置に格納されていても良い。また、ポリシールール生成計算機 151 は、記憶装置 153 と同じ筐体に格納されていても良い。

## (2) ポリシールール生成環境のソフトウェア構成

図 10 を用いて、ポリシールール生成環境のソフトウェア構成を説明する。図 10 において、ルール生成ツール 152 は、ポリシールール生成計算機 151 のプロセッサによって実行されるプログラムであり、本発明のポリシールール生成方法を実行する。ルール生成ツール 152 は、ポリシールール生成計算機 151 上で実行されているオペレーティングシステム（図示していない）に入出力の要求を行うことにより、記憶装置 153 に格納されたジョブ実行スケジュール 200、ジョブ実行条件リスト 300、アクションリスト 400、イベントリスト 500 を読み込み、またポリシールールリスト 600 を書き込むことができる。

### 【0016】

ジョブ実行スケジュール 200（図 2）、ジョブ実行条件リスト 300（図 3）、アクションリスト 400（図 4）、イベントリスト 500（図 5）、ポリシールールリスト（図 6）は、記憶装置 153 に格納されたデータであり、それぞれ別の図に示す形式で情報を格納している。

### 【0017】

本実施例では、情報処理システムのシステム管理者は、情報処理システムのポリシールールを生成しようとする場合には、あらかじめジョブ実行スケジュール 200、ジョブ実行条件リスト 300、アクションリスト 400、イベントリスト 500 を記憶装置 153 に格納し、ポリシールール生成計算機 151 においてポリシー生成ツール 152 を実行する。実行されたポリシー生成ツール 152 は、ジョブ実行スケジュール 200、ジョブ実行条件リスト 300、アクションリ

スト 400、イベントリスト 500 を読み込み、本発明のポリシールール生成方法を実行して、生成されたポリシールールを記憶装置 153 のポリシールールリスト 600 に書き込む。情報処理システムのシステム管理者は、書き込まれたポリシールールリスト 600 を読み出し、情報処理システムに適用する。

#### 【0018】

また、前記情報処理システムのシステム管理者は、例えばジョブ実行スケジュール 200 に記載されていない新しいジョブを実行する必要がある場合や、業務の都合でジョブ実行スケジュール 200 に記載された特定のジョブの実行開始時刻の変更が必要になった場合など、ジョブ実行スケジュール 200、ジョブ実行条件リスト 300、アクションリスト 400、イベントリスト 500 の記載内容のいずれかに変更が生じた場合は、再度ポリシー生成ツール 152 を実行して、新しいポリシールールリスト 600 を生成し、情報処理システムに適用する。ポリシー生成ツール 152 の再実行は、前記情報処理システムのシステム管理者が手作業で行っても、あるいはポリシールール生成ツール 152 またはそれとは別のプログラムが、記憶装置 153 に格納されたジョブ実行スケジュール 200、ジョブ実行条件リスト 300、アクションリスト 400、イベントリスト 500 の変更や、ジョブの新規受付などを検出して、自動的にポリシールール生成ツール 152 を呼び出す形で実行しても良い。

#### 【0019】

なお、本実施例において、ルール生成ツール 152 は、プログラムとして取り扱ったが、これは本発明においては、これと同等の機能がプログラムとして実装されなければならないことを意味するものではない。

#### (3) ジョブ実行スケジュール

図 2 を用いて、本実施例におけるジョブ実行スケジュール 200 の形式と内容を説明する。図 2 で、ジョブ実行スケジュール 200 は表形式になっており、1 つ以上の行 201 a, 201 b, 201 c, … から成る。すべての行は 4 つの列を含んでいる。ここで 4 つの列とは、ジョブ名 202 a, 202 b, 202 c, …, 開始時刻 203 a, 203 b, 203 c, …, 終了時刻 204 a, 204 b, 204 c, …, 計算機名 205 a, 205 b, 205 c, … である。図示して

いないが、ジョブ実行スケジュール 200 の各行は、これ以外の列を含んでも良い。

#### 【0020】

ジョブ実行スケジュール 200 に格納された情報は、ポリシールールリスト 600 を生成するポリシールール生成環境と、ジョブを実行する情報処理システムにおいて利用される。ジョブ実行スケジュール 200 に格納された情報は、情報処理システムのシステム管理者が、各ジョブの情報処理システムでの実行の要求者からの要求を受けて作成しても良いし、当該システム管理者とは別の担当者が作成したものをシステム管理者に渡し、システム管理者が使用しても良い。また、ジョブ実行スケジュール 200 に格納される情報は、システム管理者や前記担当者が手作業で作成しても、あるいは何らかのツールやユーティリティを用いて、作業の一部または全部を自動化して作成しても良い。

#### 【0021】

ジョブ実行スケジュール 200 は、情報処理システムにおいて実行する各ジョブにつき、実行開始時刻と、終了時刻と、当該ジョブを実行する情報処理システム内の計算機名が格納される。行 201 a の場合、ジョブの名前がジョブ名 202 a に、当該ジョブの開始時刻が開始時刻 203 a に、当該ジョブの終了時刻が終了時刻 204 a に、当該ジョブを実行する計算機の名前が計算機名 205 a に格納される。なお、終了時刻 204 a については、特別な場合として、例えば途中でエラーが発生し、再起動しなかった場合など、もしそのジョブを実行できなかった場合は、値として -1（マイナス 1）を格納する。行 201 b, 201 c, …についても同様である。

#### (4) ジョブ実行条件リスト

図 3 を用いて、本実施例におけるジョブ実行条件リスト 300 の形式と内容を説明する。図 3 で、ジョブ実行条件リスト 300 は表形式になっており、1 つ以上の行 301 a, 301 b, 301 c, …から成る。すべての行は 4 つの列を含んでいる。ここで 4 つの列とは、ジョブ名 302 a, 302 b, 302 c, …, 要求終了時刻 303 a, 303 b, 303 c, …, 損害額評価関数名 304 a, 304 b, 304 c, …, パラメータ 305 a, 305 b, 305 c, …である。

。図示していないが、ジョブ実行条件リスト 300 の各行は、これ以外の列を含んでいても良い。

#### 【0022】

ジョブ実行条件リスト 300 に格納された情報は、ポリシールールリスト 600 を生成するポリシールール生成環境において利用される。ジョブ実行条件リスト 300 に格納された情報は、情報処理システムのシステム管理者が、各ジョブの情報処理システムでの実行の要求者からの要求を受けて作成しても良いし、当該システム管理者とは別の担当者が作成したものをシステム管理者に渡し、システム管理者が使用しても良い。また、ジョブ実行条件リスト 300 に格納される情報は、システム管理者や前記担当者が手作業で作成しても、あるいは何らかのツールやユーティリティを用いて、作業の一部または全部を自動化して作成しても良い。

#### 【0023】

ジョブ実行条件リスト 300 は、情報処理システムにおいて実行する各ジョブにつき、その実行条件を格納している。本実施例では、ジョブの実行条件は、ジョブの要求終了時刻と、その要求終了時刻が満たされなかった場合の、損害額の計算方法を指定することによって表される。ジョブ実行条件リスト 300 の 1 つの行は、1 つのジョブに対応付けられ、そのジョブの実行条件を示す。前記 1 つの行の内容を、行 301 a を例として説明する。ジョブ名 302 a にはジョブの名前が格納され、行 301 a を特定のジョブに結びつける。要求終了時刻 303 a には、当該ジョブが終了しなければいけない時刻が格納される。損害評価関数 304 a には、当該ジョブが要求終了時刻 303 a に終了しなかった場合に、損害額を計算するための式が格納される。ジョブ名 302 a で指定されたジョブが要求終了時刻 303 a に格納された時刻に終了しなかった場合には、この式が、2 つの引数と共に実行され、損害額が計算される。ここで、2 つの引数とは、当該ジョブの終了時刻と要求終了時刻 303 a に記載された時刻との差（遅延時間）と、パラメータ 305 a に格納された値である。本実施例では遅延時間の単位を 1 時間とするが、その他の単位（分など）を単位としても良い。また、特別な場合として、もしそのジョブが実行できなかった場合は、遅延時間は -1（マイ

ナス 1) とする。なお、損害評価関数 304 a に格納される式は、呼び出すべき関数名だけが格納してあっても、何らかのプログラミング言語で記述されたサブルーチンを格納しても良い。

#### 【0024】

図3の行301 b, 301 c, …についても、行301 aと同様である。図3では、行301 bと行301 cにジョブ実行条件の具体例を示している。行301 bに記載された実行条件によれば、もしジョブAなる名称のジョブの終了時刻が、要求終了時刻の午前4時(303 bに記載されている)を過ぎて午前5時となった場合、304 bに記載された式が遅延時間の1時間とパラメータの50万を引数として呼び出され、損害額が150万と算定される。行301 cに記載された実行条件によれば、もしジョブBなる名称のジョブの終了時刻が、要求終了時刻の午前6時(303 cに記載されている)を過ぎて午前7時となった場合、304 cに記載された式が呼び出され、50万が損害額となる。なお、損害評価関数304 b, 304 cのどちらの場合も、前記の式を適用する前に、まず引数として渡された遅延時間が-1(マイナス1)かどうかを検査して、真であれば別の式を適用する処理が含まれていても良い。

#### (5) アクションリスト

図4を用いて、本実施例におけるアクションリスト400の形式と内容を説明する。図4で、アクションリスト400は表形式になっており、1つ以上の行401 a, 401 b, 401 c, …から成る。すべての行は4つの列を含んでいる。ここで4つの列とは、通番411 a, 411 b, 411 c, …, ユーザインタフェース部へのアクション412 a, 412 b, 412 c, …, 制御部へのアクション413 a, 413 b, 413 c, …, 再スケジューリング方法414 a, 414 b, 414 c, …である。図示していないが、アクションリスト400の各行は、これ以外の列を含んでいても良い。

#### 【0025】

アクションリスト400に格納された情報は、ポリシールールリスト600を生成するポリシールール生成環境と、ジョブを実行する情報処理システムにおいて利用される。アクションリスト400に格納された情報は、情報処理システム

のシステム管理者が作成しても良いし、情報処理システムの設計時に、設計者が作成しても良い。また、アクションリスト 400 に格納される情報は、システム管理者や前記設計者が手作業で作成しても、あるいは何らかのツールやユーティリティを用いて、作業の一部または全部を自動化して作成しても良い。

#### 【0026】

アクションリスト 400 は、情報処理システムにおいて障害などのイベントが発生したときに、取り得るアクションの候補がすべて列挙されている。アクションリスト 400 の 1 つの行は、1 つのアクションに対応付けられ、そのアクションの内容を示す。アクションに記載された内容は、ポリシールールリスト (図 6) において、イベントの種類と、時刻と共に指定され、情報処理システムにおいて、指定された時刻に、指定された種類のイベントが発生した場合に、実行される。前記 1 つの行の内容を、行 401 a を例として説明する。通番 411 a には、アクションリスト内で一意な通し番号が振られている。通し番号の振り方は、結果として行 401 a, 401 b, 401 c, … のそれぞれに一意な番号が振られれば、どのような方法でも良い。通番 411 a は、イベントリスト (図 5) および、ポリシールールリスト (図 6) において、特定のアクションを指定するために用いられる。ユーザインタフェース部へのアクション 412 a には、このアクションを実行するときに、ユーザインタフェース部 (図 9) において行う処理が指定される。ここでは、情報処理システムのシステム管理者に連絡するかどうか、連絡する場合どのような方法を使うか、などが指定される。ユーザインタフェース部へのアクション 412 a の具体的な格納形式としては、呼び出すべき関数名だけを格納してあっても、何らかのプログラミング言語で記述されたサブルーチンを格納しても良い。制御部へのアクション 413 a には、このアクションを実行するときに、制御部 (図 9) において行う処理が指定される。ここでは、実行中のジョブを再実行するかどうか、同じ計算機で再実行するかどうか、などが指定される。制御部へのアクション 413 a の具体的な格納形式としては、呼び出すべき関数名だけが格納してあっても、何らかのプログラミング言語で記述されたサブルーチンを格納しても良い。再スケジューリング方法 414 a には、このアクションを取った結果、ジョブ実行スケジュール 200 をどのように変更

するかが記述される。ここでは、例えば、同じ計算機でこのジョブの後に実行される予定のすべてのジョブにつき、開始時間を遅らせる、などの処理が指定される。再スケジューリング方法 414a の具体的な格納形式としては、呼び出すべき関数名だけが格納してあっても、何らかのプログラミング言語で記述されたサブルーチンを格納しても良い。

#### 【0027】

図 4 の行 401b, 401c, …についても、行 401a と同様である。図 4 では、行 401b, 行 401c に、アクションの具体例を示している。行 401b に記載されたアクションが実行された場合、ユーザインタフェース部へのアクション 412b に記載された処理により、ユーザインタフェース部（図 9）ではシステム管理者に通知するための処理は何も行わず、制御部へのアクション 413b に記載の処理により、イベントが発生したジョブを同じ計算機で即時再実行する処理を制御部（図 9）で実行し、再スケジューリング方法 414b に記載の処理により、同じ計算機でイベントが発生したジョブの後に実行される予定のすべてのジョブにつき、開始時間を遅らせるようにジョブ実行スケジュール 200 を変更する。また、行 401c に記載されたアクションが実行された場合、ユーザインタフェース部へのアクション 412c に記載された処理により、ユーザインタフェース部においてシステム管理者に通知するための処理が実行され、制御部へのアクション 413c に記載の処理により、イベントが発生したジョブを停止する処理を制御部で実行し、再スケジューリング方法 414c に記載の処理により、ジョブ実行スケジュール 200 は変更しない。

#### （6）イベントリスト

図 5 を用いて、本実施例におけるイベントリスト 500 の形式と内容を説明する。図 5 で、イベントリスト 500 は表形式になっており、1 つ以上の行 501a, 501b, 501c, …から成る。すべての行は 3 つの列を含んでいる。ここで 3 つの列とは、通番 511a, 511b, 511c, …, イベントの種類 512a, 512b, 512c, …, 取り得るアクションのリスト 513a, 513b, 513c, …である。図示していないが、イベントリスト 500 の各行は、これ以外の列を含んでいても良い。



**【0028】**

イベントリスト500には、情報処理システムにおいて発生しうる、障害などのすべてのイベントが列挙されている。イベントリスト500の1つの行は、1つのイベントに対応付けられ、そのイベントの内容を示す。イベントに記載された内容は、ポリシールールリスト（図6）において、アクションと、時刻と共に指定される。

**【0029】**

前記1つの行の内容を、行501aを例として説明する。通番511aには、イベントリスト500内で一意な通し番号が振られている。通し番号の振り方は、結果として行501a, 501b, 501c, …のそれぞれに一意な番号が振られれば、どのような方法でも良い。通番511aは、ポリシールールリスト（図6）において、特定のアクションを指定するために用いられる。障害イベントの種類512aには、行501aに対応付けられたイベントの種類が格納される。イベントの種類は、計算機がダウンした場合や、ジョブを構成するプログラムが異常終了した場合などを含んでいる。取り得るアクションのリスト513aには、行501aに記載のイベントが発生したときに、取ることができるアクションのリストが、アクションリスト400の通番のリストとして格納される。

**【0030】**

図5の行501b, 501c, …についても、行501aと同様である。図5では、行501b, 行501cに、イベントの具体例を示している。行501bに記載されたイベントは、通番511bに記載されている通り、通し番号として1が割り当てられており、障害イベントの種類512bに記載の内容により、このイベントは計算機がダウンした場合を指すと指定されている。また、取り得るアクションのリスト513bに記載の内容により、このイベントが発生した場合は、取り得るアクションはアクションリスト400の行401cに記載のアクション（ジョブを再実行せず、システム管理者に連絡する）に限られると指定されている。また、行501cに記載されたイベントは、通番511cに記載されている通り、通し番号として2が割り当てられており、障害イベントの種類512cに記載の内容により、このイベントは実行中のジョブが異常終了した場合を指

すと指定されている。また、取り得るアクションのリスト 513c に記載に内容により、このイベントが発生した場合は、取り得るアクションはアクションリスト 400 の行 401b に記載のアクション（同じ計算機で、当該ジョブを再実行し、当該ジョブの後で実行されるジョブは実行開始を遅らせる）か、行 401c に記載のアクション（ジョブを再実行せず、システム管理者に連絡する）に限られると指定されている。

### 【0031】

イベントリスト 500 に格納された情報は、ポリシールールリストを生成するポリシールール生成環境と、ジョブを実行する情報処理システムにおいて利用される。イベントリスト 500 に格納された情報は、情報処理システムのシステム管理者が作成しても良いし、情報処理システムの設計時に、設計者が作成しても良い。また、イベントリスト 500 に格納される情報は、システム管理者や前記設計者が手作業で作成しても、あるいは何らかのツールやユーティリティを用いて、作業の一部または全部を自動化して作成しても良い。

#### (7) ポリシールールリスト

図 6 を用いて、本実施例におけるポリシールールリスト 600 の形式と内容を説明する。図 6 で、ポリシールールリスト 600 は表形式になっており、1 つ以上の行 601a, 601b, 601c, … から成る。すべての行は 4 つの列を含んでいる。ここで 4 つの列とは、ジョブ名 611a, 611b, 611c, …, 時刻 612a, 612b, 612c, …, イベント 613a, 613b, 613c, …, アクション 614a, 614b, 614c, … である。図示していないが、ポリシールールリスト 600 の各行は、これ以外の列を含んでいても良い。

ポリシールールリスト 600 は、図 10 を用いてすでに説明したように、ルール生成ツール 152 によって生成される。ポリシールールリスト 600 は、ジョブを実行する情報処理システムにおいて利用される。

### 【0032】

ポリシールールリスト 600 には、「あるジョブの実行中のある時刻に、あるイベントが発生した場合に、そのジョブについてどのアクションを取るか」の組（すなわちポリシールール）が列挙される。ポリシールールリスト 600 の 1 つ

の行は、1つのポリシールールに対応付けられる。

#### 【0033】

前記1つの行の内容を、行601aを例として説明する。ジョブ名611aは、このポリシールールが対象とするジョブのジョブ名が格納される。時刻612aには、このポリシールールが有効となる時刻が格納される。時刻612aの格納形式としては、本実施例ではこのポリシールールが対象とするジョブの開始時刻から見た時間帯で格納される。ただし、時刻612aは、他の時刻の形式、例えば絶対時刻が格納されても良い。イベント613aは、このポリシールールが有効になるイベントを指定される。より具体的には、イベント613aには、イベントリスト（図5）の中の、特定のイベントの通番が格納される。アクション614aは、このポリシールールにおいて、実行されるアクションを指定する。

#### 【0034】

図6の行601b、601c、…についても、行601aと同様である。図6では、行601b、行601cに、イベントの具体例を示している。行601bに記載されたイベントは、ジョブ名611b、時刻612b、イベント613b、アクション614bに記載の内容により、「ジョブAにおいて、開始後30分以内に、ジョブAを構成するプログラムが異常終了したら、ジョブAを同じ計算機で即時再実行し、同じ計算機で実行予定の後続のジョブは実行開始を遅らせる」というポリシールールを意味する。行601cに記載されたイベントは、ジョブ名611c、時刻612c、イベント613c、アクション614cに記載の内容により、「ジョブAにおいて、開始後30分以降60分以内に、ジョブAを構成するプログラムが異常終了したら、ジョブAの実行を停止し、システム管理者に連絡する」というポリシールールを意味する。

#### （8）ポリシールールの生成方法

図7および図8を用いて、本実施例におけるポリシールールの生成方法を説明する。本実施例では、図7および図8に示すポリシールールの生成方法は、情報処理システムの管理者が、図10のルール生成ツール152を実行することによって実行される。

#### 【0035】

システム管理者から呼び出されたルール生成ツール 1 5 2 は、まず、2 つの変数 S、T の値を初期化する (7 0 1)。S の値は、非常に大きな金額とする。例えば、情報処理システムを運営している組織または情報処理システムが設置された国家の年間予算とか、資本金などを用いればよい。T の値は、情報処理システムにおいて最初のジョブが実行開始される時刻とする。最初のジョブが実行される時刻は、ジョブ実行スケジュール 2 0 0 を走査することで容易に決定することができる。

#### 【0 0 3 6】

次に、ルール生成ツール 1 5 2 は、ジョブ実行スケジュール 2 0 0 を走査し、変数 T に記載の時刻に実行中のジョブ名のリストを作成して、変数 L に格納し (7 0 2)、さらにリストになっている変数 L の最初の要素を変数 J に格納する (7 0 3)。

#### 【0 0 3 7】

次に、ルール生成ツール 1 5 2 は、イベントリスト 5 0 0 (図 5) を読み込み、イベントリスト 5 0 0 の最初の行の内容を、変数 X に格納する (7 0 4)。

#### 【0 0 3 8】

次に、ルール生成ツール 1 5 2 は、変数 X の内容 (イベントリスト 5 0 0 の 1 つの行の内容が格納されている) から、取り得るアクションのリストを取り出し、変数 M に格納し (7 0 5)、さらにリストになっている変数 M の最初の要素を取り出す。当該要素は、アクションリスト 4 0 0 (図 4) の通番の 1 つなので、ルール生成ツール 1 5 2 は、アクションリスト 4 0 0 を走査し、当該要素の通番を持つアクションリスト 4 0 0 の行を読み込んで、変数 A に格納する (7 0 6)。

#### 【0 0 3 9】

次に、ルール生成ツール 1 5 2 は、ジョブ実行スケジュール 2 0 0 を変数 C に読み込み、変数 A の内容 (アクションリスト 4 0 0 の 1 つの行、すなわち 1 つのアクション) から、再スケジュールリング方法を取り出し、取り出した再スケジュールリング方法により、変数 C に格納されたジョブ実行スケジュールを変更する (7 0 7)。例えば再スケジュールリング方法として、サブルーチン (何らかのプロ

プログラミング言語で記載されている) が格納されている場合は、変数Cの内容と、変数Jに格納されたジョブ名と、変数Tに格納された時刻とを引数として、そのプログラミング言語の処理系を呼び出して、そのサブルーチンを実行させる。変数Cの内容は、ステップ707の結果、ジョブ実行スケジュールの各行の、開始時刻、終了時刻、計算機名が、もとのジョブ実行スケジュール200の内容と変更される。特に、変数Aの内容が、ジョブ再実行をしない指定になっている場合、変数Jに格納されたジョブ名を持つ行については、終了時間が-1 (マイナス1) になっている。

#### 【0040】

次に、ルール生成ツール152は、変数Cに格納されたジョブ実行スケジュールに記載されたジョブのそれぞれが、要求終了時間を満たしているかどうか調べ、満たしていないジョブがある場合は、満たしていないことによる損害額を計算し、当該損害額の合計を計算する(708)。より具体的な損害額の計算方法は、後で図1を用いて説明する。

#### 【0041】

次に、ルール生成ツール152は、計算された損害額が、変数Sに格納された値より小さいかどうか比較する(709)。もし小さければ、計算した損害額を変数Sに格納し(710)、変数T、変数J、変数X、変数Aの内容を、変数Rに格納する(711)。もし小さくなければ、ステップ710とステップ711は実行しない。なお、ステップ711において、もし計算された損害額が0 (ゼロ) であれば、ステップ804 (図8) にジャンプしても良い。

#### 【0042】

次に、ルール生成ツール152は、変数Mに格納されたリストに、次の要素 (アクションの通番) があるかどうかを調べ(801)、もしなければ、ステップ803にジャンプする。もしあればステップ706と同様に、アクションリスト400を走査し、当該要素の通番を持つアクションリスト400の行を読み込んで、変数Aに格納し(802)、ステップ707にジャンプする。

#### 【0043】

次に、ルール生成ツール152は、変数Rに格納された内容から、ポリシール

ール（ポリシールールリスト 600 の 1 つの行）を作成し、ポリシールールリスト 600 に追加する（803）。具体的には、変数 R には、損害額が最小になる時の変数 T、変数 J、変数 X、変数 A の内容が格納されているので、変数 R 内の変数 J に相当する部分に格納されたジョブ名をポリシールールのジョブ名に、変数 R 内の変数 T に相当する部分に格納された時刻と、ジョブ実行スケジュールに格納された当該ジョブの開始時刻の差をポリシールールの時刻に、変数 R 内の変数 X に相当する部分に格納されたイベントの通番をポリシールールのイベントに、変数 R 内の変数 A に相当する部分に格納されたアクションの通番をポリシールールのアクションとして、ポリシールールを作成する。

#### 【0044】

次に、ルール生成ツール 152 は、イベントリスト 500 を読み込み、変数 X（イベントリスト 500 の行の 1 つの内容が格納されている）に格納された通番を持つ行の次の行があるかどうか調べ（804）、もしなければ、ステップ 806 にジャンプする。もしあれば、変数 X に当該次の行の内容を格納し（805）、ステップ 705 にジャンプする。

#### 【0045】

次に、ルール生成ツール 152 は、変数 L に格納されたジョブ名のリストに、変数 J に格納されたジョブ名に相当する要素の次の要素があるかどうか調べ（806）、もしなければ、ステップ 808 にジャンプする。もしあれば、変数 J に当該次の要素の内容（ジョブ名）を格納し（806）、ステップ 704 にジャンプする。

#### 【0046】

次に、ルール生成ツール 152 は、変数 T に格納された時刻を 1 単位進める（808）。時刻を進める単位は、情報処理システムのシステム管理者が、ジョブ実行スケジュール 200 を作成する際に、各ジョブの開始時刻を決める際の単位（1 分、30 分、1 時間など）を使えば良い。

#### 【0047】

次に、ルール生成ツール 152 は、ジョブ実行スケジュール 200 を走査し、変数 T に格納された時刻に実行中のジョブがあるかどうか調べ（809）、もし

あればステップ702にジャンプする。もしなければ、ルール生成ツール152を終了させる。

#### 【0048】

図7および図8に記載の方法により、ルール生成ツール152は、ある時刻（T）とその時刻に実行中のジョブ（J）と障害の種類（X）の組合せにおいて、取り得るアクションのそれぞれに対して、損害額の合計を計算し、当該損害額の合計が最も少ないアクション（A）を決定し、「時刻Tにおいて、障害Xが発生したら、アクションAを実行する」というポリシールールを、ジョブJのポリシールールとして一つ決定することができる。さらにポリシールール生成ツール152は、同じ手続きを、時刻Tとその時刻に実行中のジョブJと障害の種類Xのあらゆる組合せに対して実行することにより、すべてのジョブの、すべての時刻における、すべての種類の障害に対するポリシールールを生成し、ポリシールールリスト600に格納することができる。

#### （9）損害額の計算方法

図1を用いて、ルール生成ツール152における損害額の計算方法を説明する。本実施例では、図1に示す流れ図は、ルール生成ツール152から呼び出せるサブルーチン（損害額計算ルーチンと呼ぶ）として実装される。損害額計算ルーチン中の各変数は、図7および図8の各変数とは独立に管理されている。損害額計算ルーチンは、ルール生成ツール707（図7）の中で、変数C（変更したジョブ実行スケジュール）を引数として呼び出される。

#### 【0049】

まず、損害額計算ルーチンは、変数Sに0（ゼロ）を格納する（111）。次に、引数として渡されたジョブ実行スケジュールの先頭の行を、変数Aに格納する（112）。

#### 【0050】

次に、損害額計算ルーチンは、ジョブ実行条件リスト300を走査し、変数Aに格納されたジョブ名と同じジョブ名を持つ、ジョブ実行条件リスト300の行を取得して、変数Bに格納する（113）。

#### 【0051】

次に、損害額計算ルーチンは、変数Aに格納された終了時刻が、変数Bに格納された要求終了時刻よりも前で、かつ変数Aに格納された終了時刻が－1（マイナス1）でないかどうか調べ（114）、もし真であれば、ステップ117にジャンプする。もし偽であれば、次のステップ115を実行する。

#### 【0052】

次に、損害額計算ルーチンは、変数Aに格納された終了時刻と、変数Bに格納された要求終了時刻の差をとって遅延時間を計算する。なお、特別な場合として、変数Aに格納された終了時刻が－1（マイナス1）であれば、遅延時間は－1（マイナス1）とする。次に、損害額計算ルーチンは、変数Bに格納された損害評価関数を、計算した遅延時間と、変数Bに格納されたパラメータとを引数として呼び出す（115）。例えば損害評価関数として、サブルーチン（何らかのプログラミング言語で記載されている）が格納されている場合は、そのプログラミング言語の処理系を呼び出して、そのサブルーチンを実行させる。

#### 【0053】

次に、損害額計算ルーチンは、ステップ115で呼び出した損害評価関数の戻り値を、変数Sに加算する（116）。

次に、損害額計算ルーチンは、引数として渡されたジョブ実行スケジュールに、次の行があるかどうか調べ（117）、次の行があれば、その行の内容を変数Aに格納し（118）、ステップ113に戻る。次の行がなければ、変数Sの内容を、損害額計算ルーチンの戻り値として返し（119）、終了する。

#### 【0054】

図1に示す損害額計算ルーチンを呼び出すことにより、ルール生成ツール152は、変数Cに格納されたジョブ実行スケジュールに記載されたジョブのそれぞれが、要求終了時間を満たしているかどうか調べ、満たしていないジョブがある場合は、満たしていないことによる損害額を計算し、当該損害額の合計を計算することができる。

#### （10）情報処理システムの構成

図9を用いて、本実施例においてジョブを実行する情報処理システム900の構成を説明する。ポリシールール生成環境150（図10）において、ルール生



成ツール 152 を用いて作成されたポリシールール 600 は、情報処理システム 900 において、ジョブ実行中に障害等のイベントが発生した場合の運用管理を、自動実行するために用いられる。

#### 【0055】

図 9 の情報処理システム 900 は、運用管理計算機 901 と、ジョブ実行計算機 902 a, 902 b と、ネットワーク 903 からなる。運用管理計算機 901 は、情報処理システム内で実行されるジョブの運用管理を行う。また、ジョブ実行計算機 902 a, 902 b は、ジョブ実行スケジュール（図 2）に記載されたジョブの実行を分担して実行する。図示していないが、運用管理計算機 901 と、ジョブ実行計算機 902 a, 902 b は、それぞれ 1 つ以上のプロセッサを有し、プログラムを実行可能である。また、図 9 ではジョブ実行計算機 902 a, 902 b は 2 台の計算機であるが、本実施例では、ジョブ実行計算機は 3 台以上でも問題はない。また、運用管理計算機 901 と、ジョブ実行計算機 902 a, 902 b は、それぞれ情報処理システムのシステム管理者から、一意の計算機名を割り当てられている。なお、運用管理計算機 901 と、ジョブ実行計算機 902 a, 902 b は、そのどれか 2 つ以上が、同じ筐体に入っているいても良い。

#### 【0056】

運用管理計算機 901 と、ジョブ実行計算機 902 a, 902 b は、ネットワーク 903 に、リンク 904 を用いて接続されている。リンク 904 は、有線または無線の接続方式であり、サブネットワークを含んでいても良い。また、運用管理計算機 901 と、ジョブ実行計算機 902 a, 902 b は、それぞれ異なる方式でネットワーク 903 に接続されていても良い。

#### 【0057】

運用管理計算機 901 では、ジョブマネージャ 951 がプログラムとして実行されている。ジョブマネージャ 951 は、情報処理システム 900 内で実行されるジョブの運用管理のための処理を実行する。図示していないが、ポリシールール生成環境 150（図 10）のジョブ実行スケジュール 200、アクションリスト 400、イベントリスト 500、ポリシールール 600 は、ジョブマネージャ 951 に読み込まれ、ジョブの運用管理のために使用される。ジョブマネージャ

951は、図示していないが、ジョブ実行スケジュール200、アクションリスト400、イベントリスト500、ポリシールール600を、情報処理システム900のシステム管理者の指示により読み込むための機能を有している。

#### 【0058】

ジョブマネージャ951は、ユーザインタフェース部952と、ルールエンジン部953を含む。ユーザインタフェース部952は、情報処理システム150で障害等のイベントが発生した場合に、ポリシールールに応じてルールエンジン部953から呼び出され、情報処理システムの管理者に通知を行う。ルールエンジン部953は、ポリシールールリスト600を読み込み、エージェント954a, 954bの状態監視部からイベント通知メッセージがネットワーク903を経由して配送されるのを待ち、イベント通知メッセージを受信したら、読み込んだポリシールールリスト600を検索して配送されたイベント通知メッセージに該当するポリシールールを取り出し、当該ポリシールールに記載されているアクションを実行する。

#### 【0059】

ジョブ実行計算機902a, 902bでは、それぞれエージェント954a, 954bがプログラムとして実行されている。エージェント954a, 954bは、それぞれ状態監視部955a, 955b、および制御部956a, 956bを含む。

#### 【0060】

エージェント954a, 954bは、ジョブマネージャ951とネットワーク903を経由して通信し、ジョブマネージャ951からの指示によりジョブ実行計算機902a, 902bでジョブを実行したり、停止したり、再起動するなどの処理を実行する。また、エージェント954a, 954bは、ジョブ実行計算機902a, 902b内で、ジョブの実行中または実行の前後に障害等のイベントが発生した場合は、ジョブマネージャ951にイベント通知メッセージを送信する。図示していないが、イベント通知メッセージは、イベントが発生したジョブ実行計算機の計算機名（前述の通り、ジョブ実行計算機のそれぞれには一意な計算機名が割り当てられている）、イベントが発生した時刻、およびイベントの

種類が含まれている。ここで、前記イベントの種類は、イベントリスト 500 に記載されたイベントの種類のうちのいずれかである。

#### 【0061】

より具体的には、エージェント 954 a の状態監視部 955 a は、ジョブ実行計算機 902 a 内で障害等のイベントが発生した場合は、ジョブマネージャ 951 にイベント通知メッセージを送信する。また、制御部 956 a は、ジョブマネージャ 951 からネットワーク経由で送信されてきた指示を受信し、その指示に応じて、指定されたジョブをジョブ実行計算機 902 a のプロセッサを用いて実行開始させたり、ジョブ実行計算機 902 a で実行中のジョブを停止したり、指定されたジョブをジョブ実行計算機 902 a で再実行したりといった、ジョブ運用管理のための処理を実行する。エージェント 954 b と、エージェント 954 b に含まれる状態監視部 955 b、制御部 956 b についても、機能はそれぞれエージェント 954 a、状態監視部 955 a、制御部 956 a と同様である。

#### 【0062】

なお、本実施例ではジョブマネージャ 951、エージェント 954 a、954 b はプログラムであるとしたが、これらの一部または全部がプログラムなくはいけないことを意味するわけではない。例えば、ジョブ実行計算機 902 a のダウンを検出して、イベント通知メッセージをジョブマネージャ 951 に送信するため、ジョブ実行計算機 902 a がダウンしていても、前記イベント通知メッセージが送信できるよう、エージェント 954 a の状態監視部 955 a の一部または全部が、ジョブ実行計算機 902 a に何らかの方式で接続（ネットワーク 903 経由でも良い）されたハードウェアとして実現されていても良い。

##### （11）情報処理システムにおけるポリシーールの利用

図 9 を用いて、情報処理システム 900 において、ジョブマネージャ 951 のルールエンジン部 953 で読み込まれるポリシーールリスト 600 が、どのように使われるかを示す。なお、前述の通り、ジョブマネージャ 951 は、ジョブ実行スケジュール 200、アクションリスト 400、イベントリスト 500 も読み込んでおり、そのためルールエンジン部 953 は、ジョブ実行スケジュール 200、アクションリスト 400、イベントリスト 500 の内容も参照できる。

**【0063】**

通常、情報処理システム900では、ジョブマネージャ951が、ジョブ実行スケジュール200を参照して、エージェント954aやエージェント954bと通信して、ジョブ実行スケジュールに記載された開始時刻に、指定された計算機名の計算機において、指定されたジョブの実行を指示し、複数のジョブが実行される。

**【0064】**

このような状況で、情報処理システム900内で、ジョブを構成するプログラムが異常終了するなどの障害を含むイベントが発生すると、まずそのジョブを実行していたジョブ実行計算機で実行されているエージェントの状態監視部がそれを検出し、前記の通り、イベントが発生したジョブ実行計算機の計算機名、イベントが発生した時刻、およびイベントの種類が含むイベント通知メッセージを作成し、ジョブマネージャ951に送信する。ジョブマネージャ951のルールエンジン部953は、当該イベント通知メッセージを受信し、当該イベント通知メッセージからイベントが発生したジョブ実行計算機の計算機名とイベントが発生した時刻を取得し、これを用いてジョブ実行スケジュールを検索することにより、イベントが発生したジョブのジョブ名を特定する。なお、当該ジョブ名は、エージェントから送信される時点で、イベント通知メッセージに含まれていても良い。この場合はジョブ実行スケジュールを検索する処理は不要である。

**【0065】**

次に、ルールエンジン部953は、イベント通知メッセージに格納されているイベントの種類を取得し、これを用いてイベントリストを検索し、該当するイベントの種類を持つ行を取得し、当該行に格納された通番を取得する。

**【0066】**

次に、ルールエンジン部953は、取得したジョブ名と、イベントの発生時刻と、通番を用いて、ポリシールールリストを検索し、該当するポリシールール（ポリシールールリスト600（図6）に記載されたポリシールールのいずれか）を取得し、そのポリシールールに格納されたアクションの通番を取得する。さらにルールエンジン部953は、取得したアクションの通番を用いて、アクション

リストを検索し、アクションの通番と同じ通番を持つ行（すなわちアクション）を取得する。

#### 【0067】

次に、ルールエンジン部 953 は、取得したアクションに記載された、ユーザインタフェース部へのアクションおよび制御部へのアクションを実行する。例えばユーザインタフェース部へのアクションおよび制御部へのアクションが、何らかのプログラミング言語によりサブルーチンとして記述されている場合は、そのプログラミング言語の処理系を呼び出し、ユーザインタフェース部へのアクションおよび制御部へのアクションに記載されている内容を実行させる。通常、ユーザインタフェース部へのアクションを実行した結果、ユーザインタフェース部 952 が呼び出され、端末へのエラーメッセージの表示や、アラームの始動など、イベントの発生を情報処理システムのシステム管理者に通知するための処理が、自動的に実行される。また、制御部へのアクションを実行した結果、制御部 956 a または制御部 956 b のどちらか、またはすべてに対してジョブマネージャから指示のメッセージがネットワーク 903 を経由して送信され、その結果としてジョブの再実行や、ジョブの別の計算機での再実行や、ジョブの停止などの運用管理処理が、自動的に実行される。

#### 【0068】

##### 【発明の効果】

本発明においては、ジョブごとに設定されるポリシールールは、各ジョブの投入者ではなく、情報処理システムのシステム管理者によって決定され、常に情報処理システム全体の損害額が最小になるようなポリシールールが設定される。従って、ジョブマネージャは、本発明によって生成されたポリシールールを用いて、情報処理システム全体として最適になるように、自動運用管理を行うことができる。

特に、本発明においては、ジョブごとに設定されるポリシールールは、各ジョブの投入者ではなく、情報処理システムのシステム管理者によって決定される。また、実行予定のジョブごとに、イベントが発生する可能性のある時刻とイベントの種類の可能なあらゆる組合せにつき、常に情報処理システム全体の損害額が最

小になるようなアクションが決定され、ポリシールールとして設定される。従って、ジョブマネージャは、本発明によって生成されたポリシールールを用いて、障害等のイベントの発生時刻やイベントの種類を加味した、情報処理システム全体として常に損害額が最小となるような、きめ細かくかつ最適な自動運用管理を行うことができる。

**【図面の簡単な説明】**

**【図 1】**

本発明の実施例における、損害額の計算方法を示す流れ図である。

**【図 2】**

本発明の実施例における、ジョブ実行スケジュール 200 の構造と内容を示す図である。

**【図 3】**

本発明の実施例における、ジョブ実行条件 300 の構造と内容および具体例を示す図である。

**【図 4】**

本発明の実施例における、アクションリスト 400 の構造と内容および具体例を示す図である。

**【図 5】**

本発明の実施例における、イベントリスト 500 の構造と内容および具体例を示す図である。

**【図 6】**

本発明の実施例における、ポリシールールリスト 600 の構造と内容および具体例を示す図である。

**【図 7】**

本発明の実施例における、ポリシールール作成方法を示す流れ図である。流れ図は図 8 に続いている。

**【図 8】**

図 7 から続く、本発明の実施例における、ポリシールール作成方法を示す流れ図である。

**【図 9】**

本発明の実施例における、ジョブを実行する情報処理システムの構成を示す図である。

**【図 1 0】**

本発明の実施例における、ポリシールールリストを生成するポリシールール生成環境の構成を示す図である。

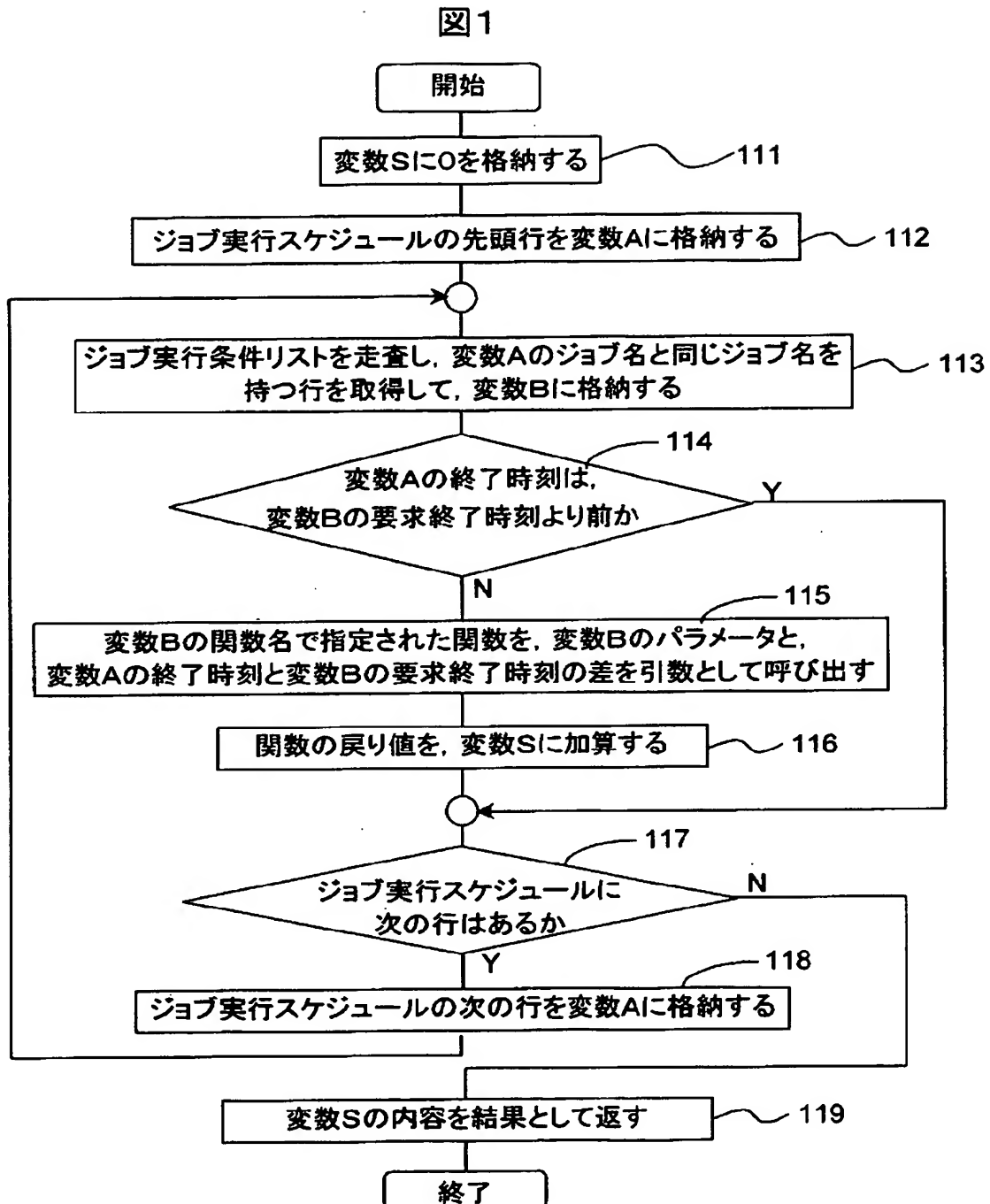
**【符号の説明】**

- 1 5 0 ポリシールール生成環境
- 1 5 2 ルール生成ツール
- 2 0 0 ジョブ実行スケジュール
- 3 0 0 ジョブ実行条件
- 4 0 0 アクションリスト
- 5 0 0 イベントリスト
- 6 0 0 ポリシールールリスト
- 9 0 0 情報処理システム
- 9 5 1 ジョブマネージャ
- 9 5 2 ユーザインタフェース部
- 9 5 3 ルールエンジン部
- 9 5 4 a, 9 5 4 b エージェント
- 9 5 5 a, 9 5 5 b 状態監視部
- 9 5 6 a, 9 5 6 b 制御部。

【書類名】

図面

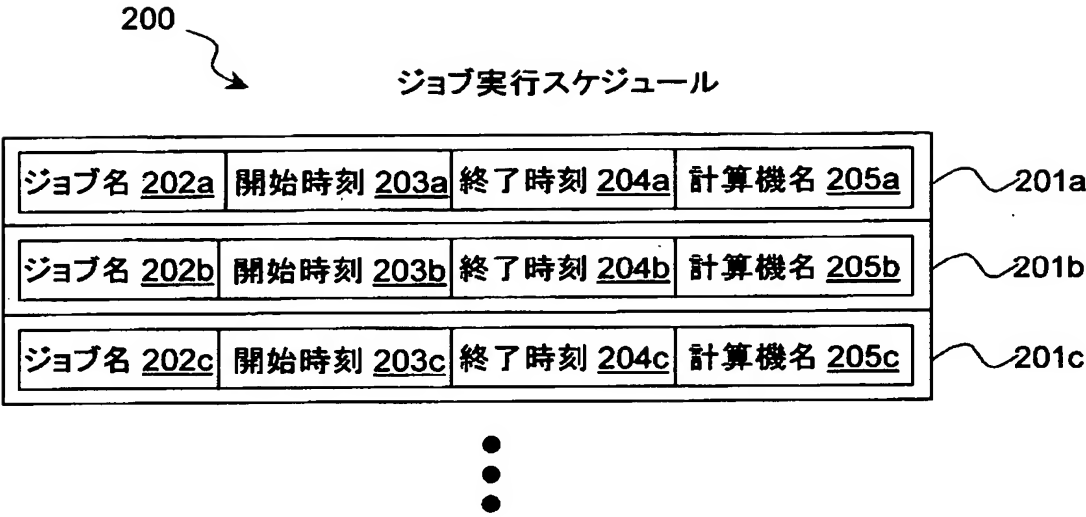
【図 1】





【図 2】

図 2



【図 3】

図3

300 ジョブ実行条件リスト

301a	302a	303a	304a	305a
ジョブ名	要求終了時刻	損害評価関数	パラメータ	

301b	302b	303b	304b	305b
「ジョブ A」	「04:00 AM」	「遅延時間に100万を乗じ、パラメータを加えて損害額とする」	「50万」	

301c	302c	303c	304c	305c
「ジョブ B」	「06:00 AM」	「遅延時間に関係なく、パラメータを損害額とする」	「50万」	

●  
●  
●

【図 4】

図4

400

アクションリスト

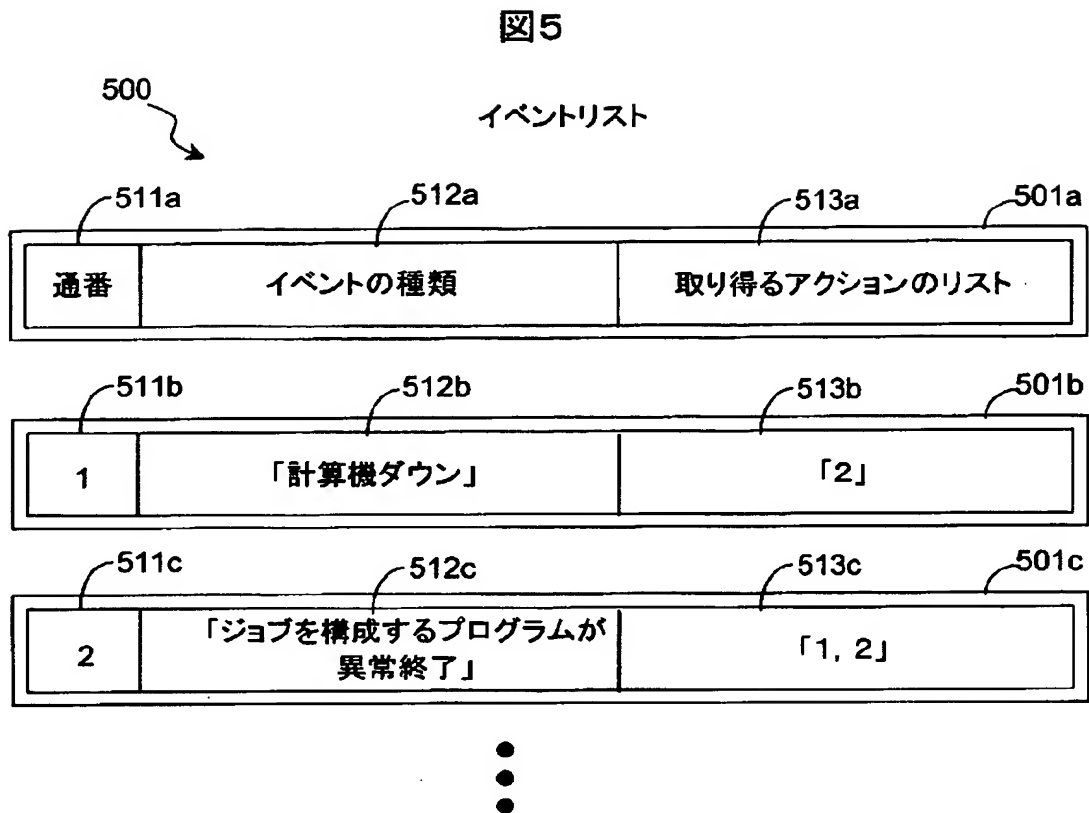
411a	412a	413a	414a	401a
通番	ユーザインタフェース部へのアクション	制御部へのアクション	再スケジューリング方法	

411b	412b	413b	414b	401b
1	「何もしない」	「同じ計算機で即時再実行する」	「同じ計算機で実行予定のジョブは、このジョブの再実行後に実行」	

411c	412c	413c	414c	401c
2	「端末にメッセージを表示して管理者に通知」	「停止させる」	「スケジュールは変更しない」	

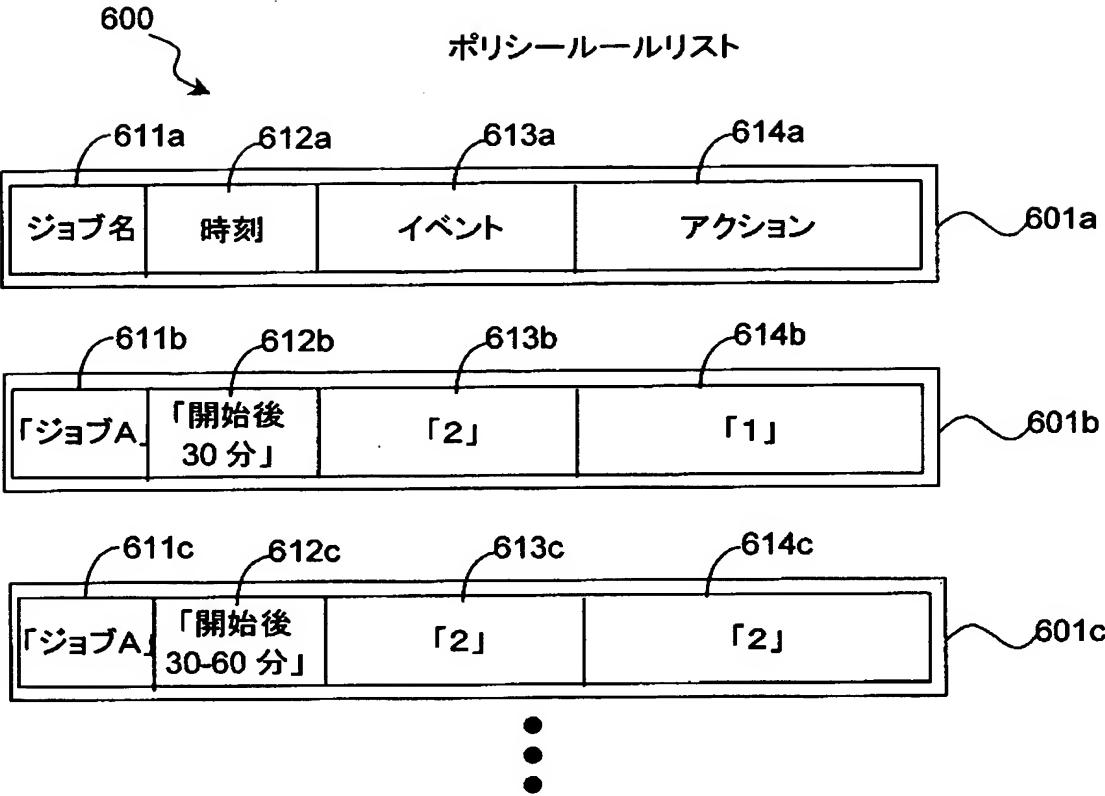
●  
●  
●

【図 5】

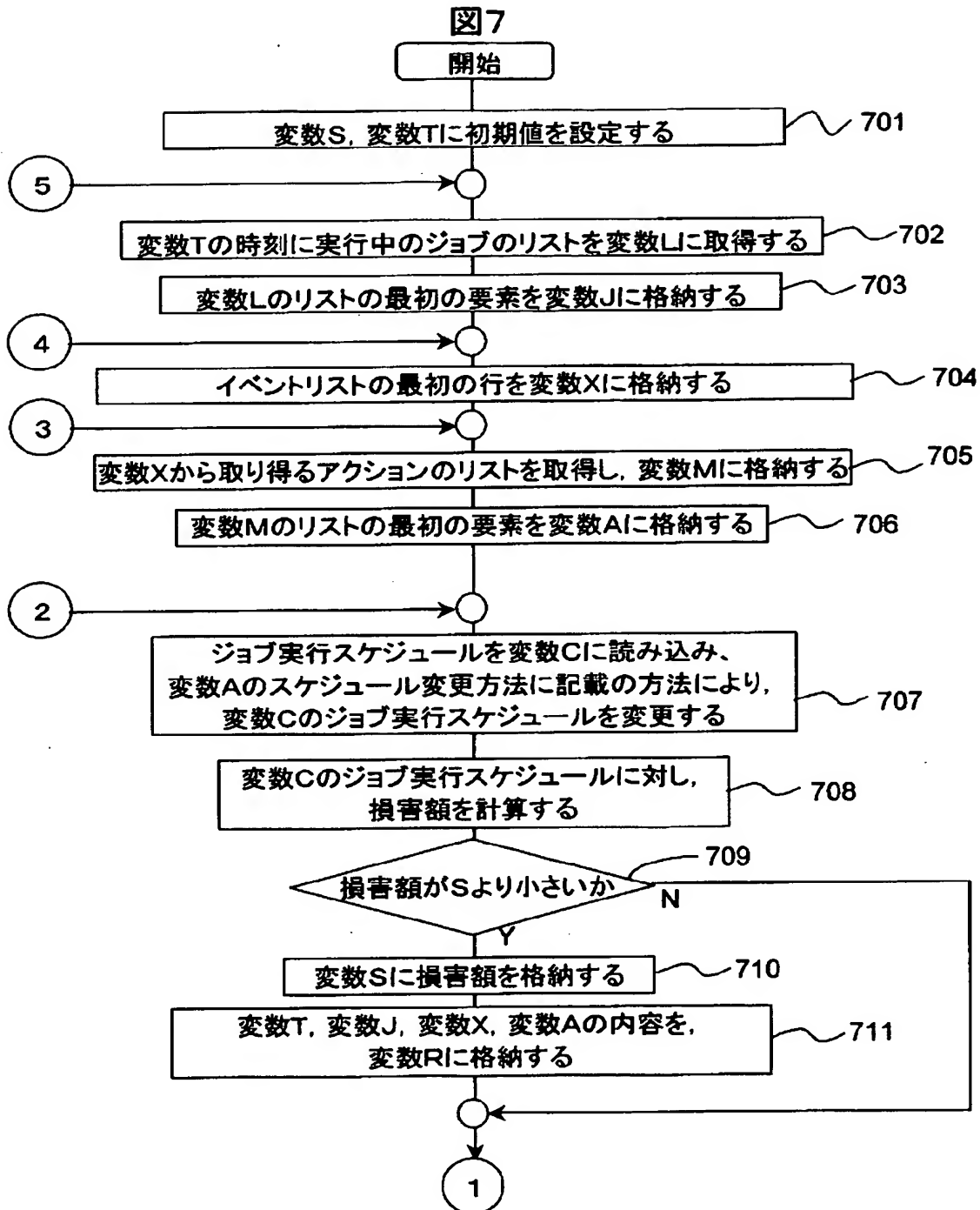


【図 6】

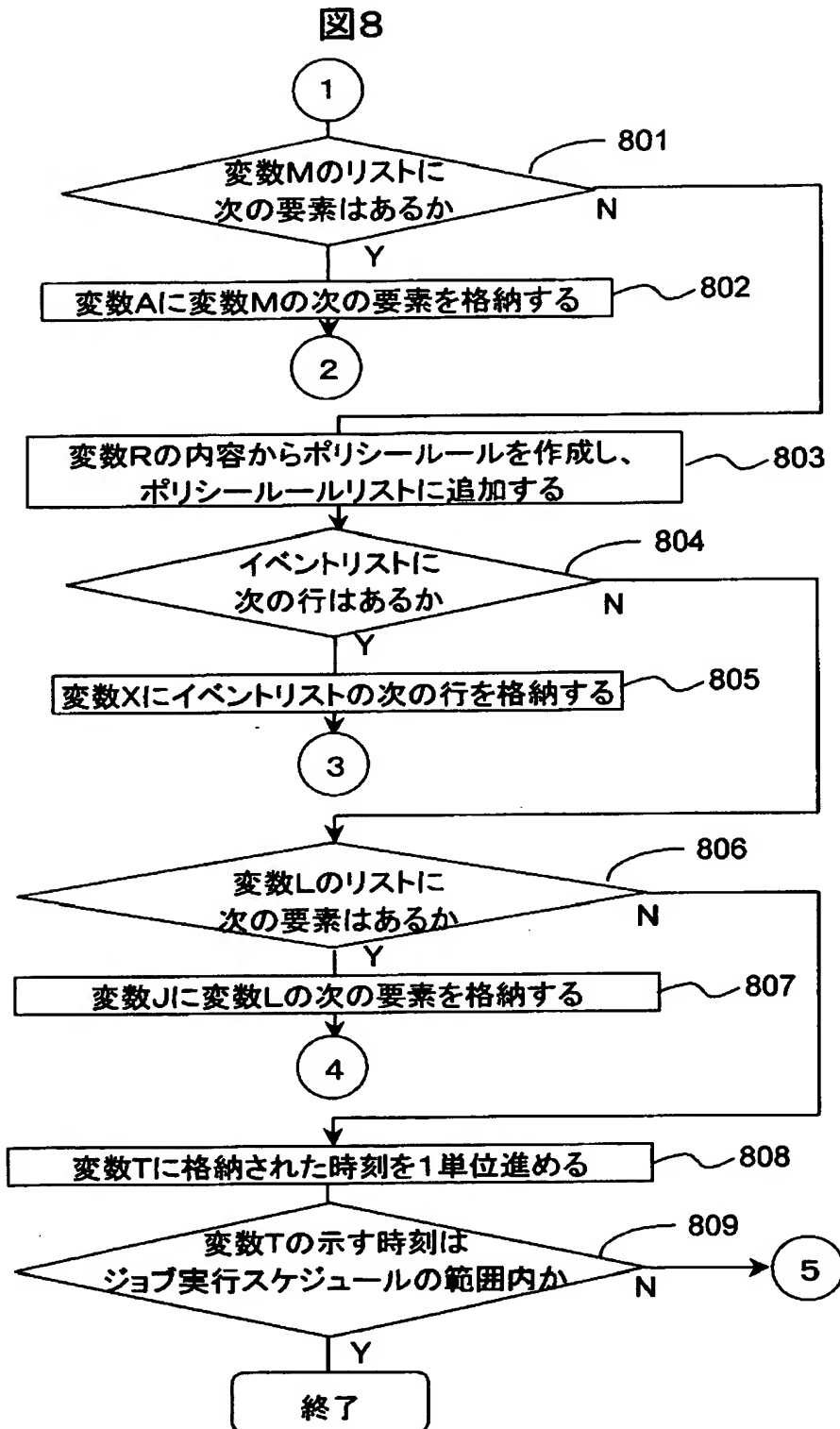
図6



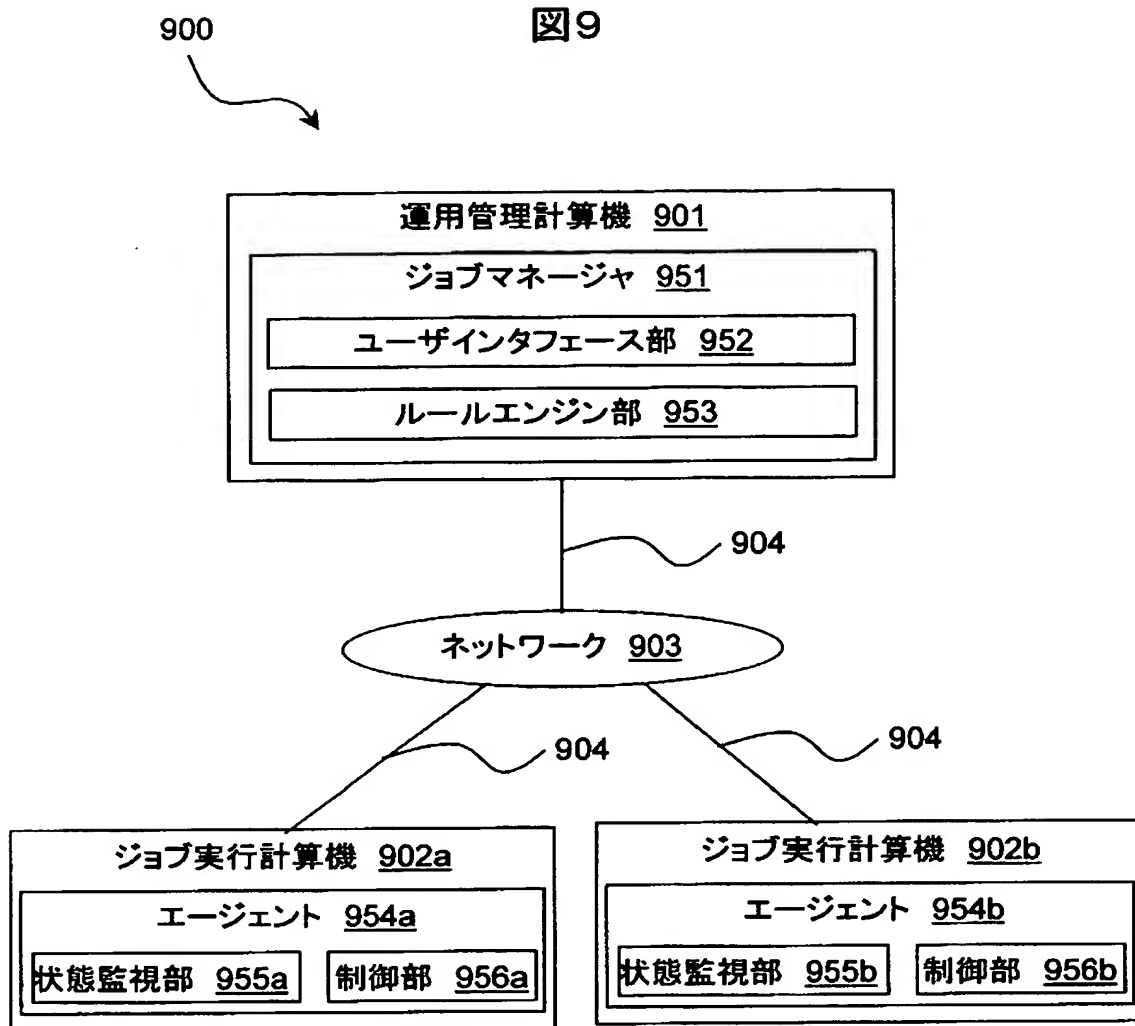
【図 7】



【図 8】



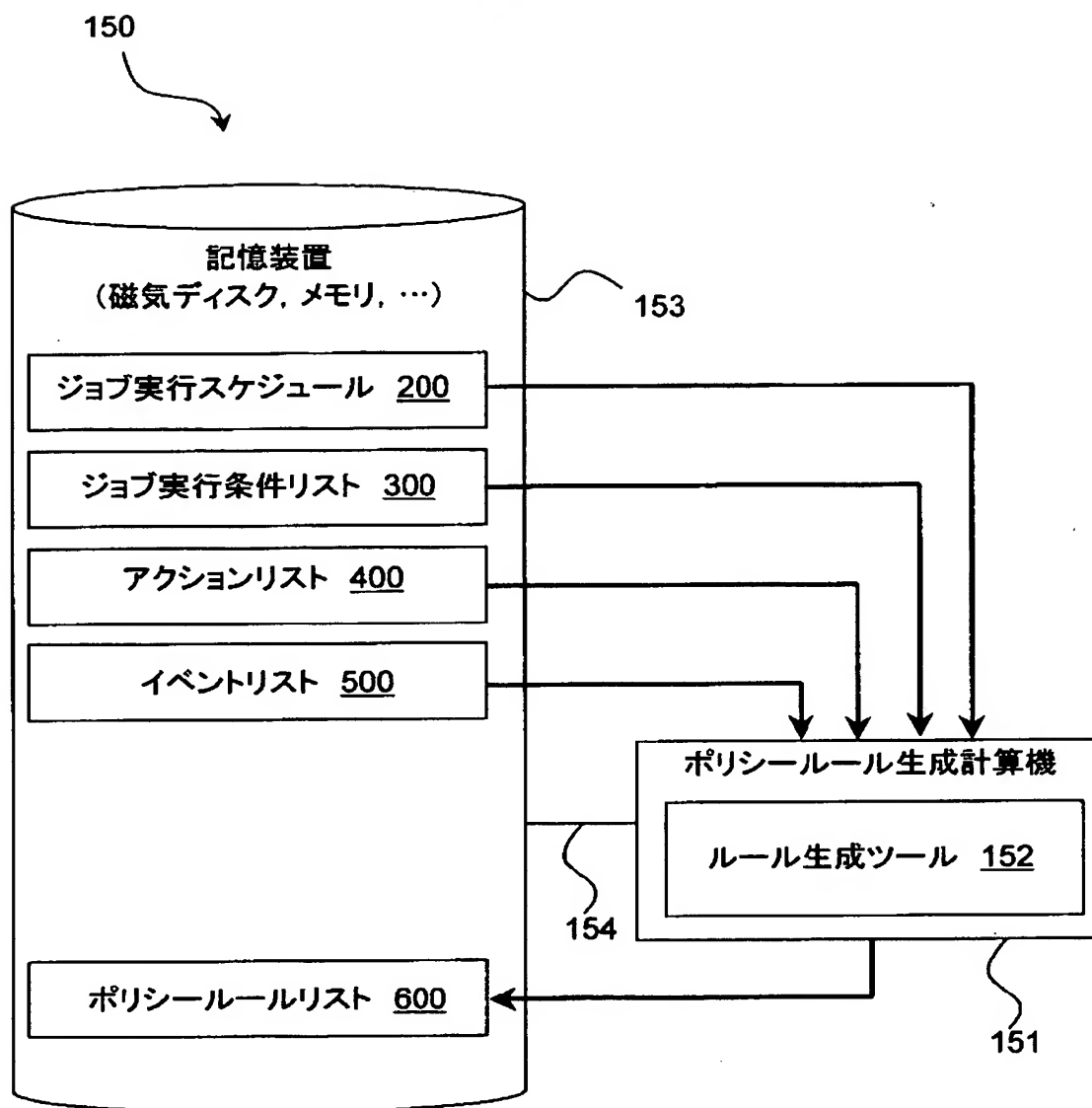
【図 9】





【図10】

図10



【書類名】 要約書

【要約】

【課題】 ポリシールールを用いてジョブ実行の運用管理を実行する情報処理システムにおいて、当該情報処理システムに最適なポリシールールを生成する方法を提供する。

【解決手段】 本発明においては、ジョブごとに設定されるポリシールールを、各ジョブの投入者ではなく、情報処理システムのシステム管理者によって決定する。また、実行予定のジョブごとに、イベントが発生する可能性のある時刻とイベントの種類可能なあらゆる組合せにつき、常に情報処理システム全体の損害額が最小になるようなアクションが決定され、ポリシールールとして設定する。

【効果】 障害等のイベントの発生時刻やイベントの種類も加味して、情報処理システム全体として常に損害額が最小となるような、きめ細かくかつ最適な自動運用管理を行うポリシールールを生成することができる。

【選択図】 図 1

認定・付加情報

特許出願の番号	特願 2003-086919
受付番号	50300500734
書類名	特許願
担当官	第七担当上席 0096
作成日	平成15年 3月28日

<認定情報・付加情報>

【提出日】	平成15年 3月27日
-------	-------------

次頁無

特願 2 0 0 3 - 0 8 6 9 1 9

出 願 人 履 歴 情 報

識別番号

[ 0 0 0 0 0 5 1 0 8 ]

1. 変更年月日

1 9 9 0 年 8 月 3 1 日

[変更理由]

新規登録

住 所

東京都千代田区神田駿河台 4 丁目 6 番地

氏 名

株式会社日立製作所